Állatok felismerése képeken

Az alkalmazás egy állatfelismerő program, amely egyszerű és felhasználóbarát módon működik. A felhasználó feltölthet egy képet a számítógépéről (JPG, JPEG vagy PNG formátumban) a program felületén keresztül. Miután a kép feltöltésre került, az alkalmazás automatikusan elemzi azt az előre betanított YOLOv8 modell segítségével. Ez a modell képes felismerni a képen található objektumokat, különösen az állatokat. Az elemzés után a program megjeleníti a feltöltött képet úgy, hogy azon a felismert állatokat és azok körvonalait jelöli ki. Az alkalmazás listában felsorolja az összes felismert állat nevét és az azonosítás pontosságát százalékos formában.

A képen szöveg, képernyőkép, Multimédiás szoftver, emlős látható

Automatikusan generált leírás

A kód megírásához a következő csomagokat használtuk: Streamlit az egyszerű és interaktív webalkalmazás készítéséhez, OpenCV a képek és a videók feldolgozásához, NumPy az adatkezeléshez, valamint az Ultralytics YOLO könyvtárat az állatok valós idejű felismerésére és detektálására.

Ultralytics YOLO

Az Ultralytics YOLO a gépi látás egyik legnépszerűbb algoritmusa, amelyet elsősorban valós idejű objektumdetekcióra fejlesztettek ki. A YOLO, vagyis a "You Only Look Once" név arra utal, hogy az algoritmus egyetlen lépésben képes azonosítani és lokalizálni a képen található objektumokat. Ez az egyedi megközelítés gyorsabbá és hatékonyabbá teszi a hagyományos módszerekhez képest, amelyek több lépésben dolgozzák fel a képeket.

A YOLO működése rendkívül egyszerű, de mégis hatékony. Az algoritmus a bemeneti képet egy rácshálóba osztja, majd minden rácselem megvizsgálja, hogy tartalmaz-e objektumot. Ha igen, az algoritmus meghatározza az objektum helyét a képen, méretét, és egy valószínűségi értéket rendel hozzá, amely azt mutatja, mennyire biztos abban, hogy az adott objektum melyik kategóriába tartozik. Az algoritmus egyetlen lépésben dolgozza fel a teljes képet, így elkerülve a régiók kijelölését és azok külön osztályozását, ami más módszerekben gyakori. Ez a megközelítés rendkívül gyors, ezért valós idejű alkalmazásokhoz, például közlekedésfigyeléshez vagy ipari gyártósorok ellenőrzéséhez is kiválóan alkalmas.

Az Ultralytics YOLO nagy előnye, hogy felhasználóbarát és könnyen alkalmazható. A Python-alapú implementáció egyszerűsíti a modell használatát, legyen szó előre betanított modellekről vagy saját adathalmazra történő újratanításról. Az Ultralytics különböző méretű modelleket kínál, amelyek között az igényekhez és a hardverkapacitáshoz igazodva lehet választani. Például a kisebb modellek gyorsak és kevés erőforrást igényelnek, míg a nagyobb modellek nagy pontosságot nyújtanak összetettebb feladatok esetén.

A yolov8n.pt egy előre betanított YOLOv8 modell súlyfájlját tartalmazza, amelyet az Ultralytics csapata hozott létre. A fájl a YOLOv8 Nano változatát képviseli, amely a legkisebb és leggyorsabb modell az YOLOv8 verziók között. A modell általános célú objektumdetekcióra van optimalizálva, és több ezer képen tanították be a COCO adathalmazon, amely számos kategóriát tartalmaz, például embereket, járműveket és állatokat.

A **COCO** (Common Objects in Context) egy nagy és széleskörű adatbázis, amelyet a képfelismerési és számítógépes látás feladatokhoz használnak, és tartalmaz képeket, amelyek különböző tárgyakat ábrázolnak különböző kontextusokban. A COCO adatbázis különösen fontos a **tárgyfelismerés** és **tárgylokalizáció** feladatokban, és a mély tanulási modellek, mint például a **YOLO** számára is széles körben használják.

A COCO adatbázis főbb részei:

* **Tárgykategóriák**: A COCO adatbázis 80 különböző tárgykategóriát tartalmaz, például emberek, állatok, járművek, háztartási eszközök, stb.
* **Annotációk**: Minden képhez többféle annotáció tartozik:
  + - **Bounding box-ok**: A képen található tárgyak pontos helyét jelölik.
    - **Osztálycímkék**: Minden tárgyhoz hozzárendelt osztály (pl. "kutya", "autó", "szék").
    - **Maszkok**: A COCO szegmentálási adatait is tartalmazza, ahol a tárgyak pixel-szintű maszkjai is elérhetőek.
    - **Feliratok**: Képhez tartozó leírások és címkék, amelyek segítik a képek jobb megértését.
* **Adatok**: A COCO adatbázis több mint 300.000 képet és több mint 2,5 millió annotációt tartalmaz, így óriási és gazdag adatforrás a gépi tanulás számára.

Programkód:



Betölti a YOLOv8 Nano előre betanított modellt a yolov8n.pt súlyfájlból. Ez az objektum képes lesz képeken objektumokat detektálni, azokat osztályozni és vizualizálni.



st.file\_uploader: Lehetővé teszi a felhasználónak, hogy képet töltsön fel az alkalmazásba. Csak jpg, jpeg és png formátumú fájlokat fogad el.

Az eredmény egy feltöltött fájlobjektum, vagy None, ha nem töltöttek fel képet.

A képen képernyőkép, szöveg, Betűtípus, sor látható

Automatikusan generált leírás

if uploaded\_file is not None: Ellenőrzi, hogy a felhasználó feltöltött-e képet.

uploaded\_file.read(): Beolvassa a feltöltött fájl bináris tartalmát.

np.asarray(bytearray(...)): Átalakítja a bináris adatokat egy NumPy tömbbé.

cv2.imdecode: A NumPy tömböt egy képpé alakítja át OpenCV formátumban.



A betöltött YOLOv8 modell (YOLOv8n) segítségével elemzi a feltöltött képet.

A model(image) metódus visszaad egy eredményobjektumot, amely tartalmazza a detektált objektumokat, azok koordinátáit, osztályait és valószínűségi értékeit.



results[0].plot(): Létrehoz egy képet, amelyen a detektált objektumok keretekkel és címkékkel vannak megjelölve.

st.image: Megjeleníti az annotált képet a Streamlit alkalmazásban. A channels="BGR" paraméter biztosítja, hogy a színek helyesen jelenjenek.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

for box in results[0].boxes: Végigmegy a detektált objektumokat tartalmazó dobozokon.

box.cls[0]: Az objektum osztályának azonosítója.

box.conf[0]: Az algoritmus által becsült valószínűségi érték, amely az adott detekció pontosságát mutatja.

model.names[cls\_id]: Az osztály ID-jának megfelelő objektumnév.

Ezután kiírja az objektum nevét és a detekció valószínűségét százalékos formában.

Források:

<https://docs.ultralytics.com/models/yolov8/>

<https://docs.ultralytics.com/datasets/detect/coco/>